Due on February 11, 2019
53 points total

General directions: We will exclusively use Python 3 for our programming assignments, and allow only the use of modules in the Python 3 standard library unless explicitly specified otherwise on an individual assignment basis. This forbids the use of common third-party libraries such as Numpy, Sympy, etc., but not the use of math or io.

Unless specified otherwise, for the X-th homework, download the single "hwX_skeleton.py" file from the course website, and rename it to "hwX.py" on your machine. When you are done and ready to submit, upload your file named **exactly** "hwX.py" on Gradescope for assignment "HW X (Programming)." When you upload your file, the autograder will run a simple test for each function so that you can confirm it was properly uploaded and executed. Generally, if an assignment involves printing or writing a file in a specific format, there will be at least one simple test that checks your output is formatted as we expect. These tests are not worth any credit — once the due date is over, your submission will be graded by a collection of additional test cases.

All answers to non-programming questions must be typed, preferably using LaTeX. If you are unfamiliar with LaTeX, you are strongly encouraged to learn it. However, answers typed in other text processing software and properly converted to a PDF file will also be accepted. To submit your file, upload your PDF on Gradescope for assignment "HW X (PDF)." Handwritten answers or PDF files that cannot be opened will not be graded and will not receive any credit.

Finally, please read the detailed collaboration policy given on the course website. You are **not** allowed to discuss homework problems in groups of more than 3 students. **Failure to adhere to these guidelines will be promptly reported to the relevant authority without exception.**

Point values: Every problem has a specified amount of points which are awarded for the correctness of your solutions. In addition, each proof-oriented problem has an additional **style point**. In the homework handout, this is signified by a "+1" in the point value. To earn this point, your solutions should be clear, well organized, and easy to follow. This is to encourage not only perfectly correct solutions, but well presented ones.

Problem 1 (10+1 points)

(a) [5 points] Express the following as a predicate formula.

There exists an integer q for which $q + x = x$ for every real number x.

(b) [5 points] Express the following as a predicate formula using the specified domain of discourse and predicates.

If n is an integer greater than 2, then there are no non-zero integer values of a, b, c that satisfy $a^n + b^n = c^n$. The domain is the set of integers and the predicates are $P(x, y, z, n) \Leftrightarrow x^n + y^n = z^n$, $Q(x) \Leftrightarrow x > 2$, and $O(x) \Leftrightarrow x = 0$.

Problem 2 (15+1 points)

For the following pairs of formulas F and G, fill in the blanks of G with quantifiers ($\forall$ or $\exists$) so that G is equivialent to F, if possible. If this is not possible, proceed onto the following.

Show that the blanks of G can be filled in so that G is implied by F, if possible. Then also show that the blanks of G can be filled in so that G implies F, if possible. If neither of these are possible, state this as your answer.

For part (a), your answers (if any) must pick at least one quantifier differently for the variables in G than in F. For parts (b) and (c), there are no restrictions on which quantifiers you may use.

(a) [5 points]

$$F = \forall x \exists y.\, P(x, y)$$
$$G = \_x\_y.\, P(x, y)$$

(b) [5 points]

$$F = \exists x \forall y.\, P(x, y)$$
$$G = \_y\_x.\, P(x, y)$$

(c) [5 points]

$$F = \forall w \forall x \exists y \exists z.\, P(w, x, y, z)$$
$$G = \_x\_w\_z\_y.\, P(w, x, y, z)$$

Problem 3 (10+1 points)

Prove or disprove for any sets A, B, C:

(a) [5 points]

$$(A \setminus B) \setminus C = (A \setminus C) \setminus (B \setminus C)$$

(b) [5 points]

$$(C \setminus (A \cup B)) \cup (B \cap C) \cup (A \cap C) = C$$

Problem 4 (15 points)

Recall that any logical formula can be expressed in disjunctive normal form (DNF). For this problem you will implement the function `convertToDNF(n,table)` as specified in the skeleton file, hw3_skeleton.py, on the course website.

You may define additional methods that are called by `convertToDNF(n,table)` as you see fit, and use the included methods `generateTable(n,f)` and `f_test(xs)` to test your implementation. The specifications and usage of all methods mentioned are described in hw3_skeleton.py.